DT12 Rec'd PCT/PTO 22 FEB 2005

BAKER BOTTS L.L.P.

30 ROCKEFELLER PLAZA

NEW YORK, NEW YORK 10112-4498

TO ALL WHOM IT MAY CONCERN:

Be it known that WE, JENS DAVID and JENS BRETSCHNEIDER, citizens of the Federal Republic of Germany, whose post office addresses are Jaegerstraße 25, D-01099 Dresden, Federal Republic of Germany; and Wittenberger Straße 22, D-01309, Dresden, Federal Republic of Germany, respectively, have invented an improvement in

METHOD FOR INITIALIZING PROGRAMMABLE SYSTEMS

of which the following is a

SUBSTITUTE SPECIFICATION

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of International Patent Application No. PCT/DE2003/001747 filed May 28, 2003, which claims priority to German Patent Application No. 102 40 770.3 filed August 30, 2002, both of which applications are hereby incorporated by reference in their entireties herein.

FIELD OF THE INVENTION

[0002] The invention relates to the operation of programming devices or systems such as computers, processors and microprocessors. In particular, the invention relates to methods for initializing such devices and systems for operation.

BACKGROUND OF THE INVENTION

[0003] The ~~invention relates to a method for~~ initializing **processes of** programmable systems, ~~in which the information required for initializing~~**for example, at power-up or start-up, involve initialization, resetting, or loading of** registers and internal and/or external modules ~~is contained~~**with start-up information. The relevant start-up information is often stored** in an external memory and is read **from the external memory at start-up**, particularly ~~for application~~ in **applications that involve** programmable system-on-chip ~~ASIC~~ elements **(e.g., Application Specific Integrated Circuits,"ASIC")**.

[0004] Complex microcontroller-assisted electronic systems~~, particularly in the region of the personnel computer periphery, are normally~~ **which use, for example, peripheral devices to a personal computer ("PC") or the like, are usually** made up of ~~few~~**a limited number of** components for production-related and economic reasons. Generally, attempts are made to accommodate all of the logic on one application-specific, integrated chip ~~(ASIC = Application Specific Integrated Circuit)~~**or ASIC**. This ASIC contains the interfaces which set up the connection to the PC, for example PCI (Peripheral Components Interconnect), ~~Cardbus~~**Card bus** controller or USB (Universal Serial Bus) controller. The nature of the products, which are produced in large quantities at low prices, means that it is usual for semiconductor design companies to develop a hard-wired solution which is produced by parties to an agreement and is sold to OEM (Original Equipment Manufacturer) customers, who integrate the semiconductors in their peripheral devices. In this case, the resultant end products usually differ in terms of scope

of functions, performance, power consumption and price, depending on what circuitry and additional components have been used for the respective solution.

[0005] These differences in circuitry mean that the various hardware devices need to be addressed and handled differently in terms of software. Operating-system drivers adapted thereto are required and are often also desired by the OEM customers in order to distinguish themselves more from competitors' products as a result.

[0006] In addition, the additional electronics on the board need to be put into an initial state in various ways when they are actually turned on or plugged in.

[0007] Two problem situations arise from the demands described:

~~First~~**Firstly**, identification needs to be performed for each external device and an appropriate device driver needs to be allocated. Modern PC operating systems need to be able to identify external modules added during operation (e.g. USB, ~~Cardbus~~**Card bus**) or before operation (e.g. PCI) automatically and to be able to allocate a suitable driver. If the features which the operating system can use for identification are accommodated exclusively on the ASIC, however, then distinction between various OEM products is not possible.

[0008] Secondly, the OEM-specific elements need to be initialized when turning on. A situation may arise in which the external electronics modules located on the device beside the ASIC need to be put into a defined initial state directly upon turning on or plugging in or within a very short time afterwards, for example in order to avoid destruction, excessive power consumption or confusing status displays, by light-emitting diodes or a display. Since the ASIC is usually the only "intelligent" chip in the circuit which can perform this task, but the external

circuitry generally differs, as described above, it is necessary to find a way of letting the ASIC have the information about the initialization of the other modules so that it can perform the initialization accordingly.

[0009] Known methods are based on accommodating a small EEPROM (Electrically Erasable Programmable Read Only Memory) on the device's board and connecting it to the ASIC ~~via a~~ usually **via a** serial bus. The EEPROM stores identification features and serial numbers. In the case of a PCI/~~Cardbus~~**Card bus**, for example, the identification features are understood to mean the product ID, vendor ID, subsystem ID and subsystem vendor ID as a revision identifier and device class. An example of a serial number is the MAC-ID in the case of Ethernet-network cards.

[0010] This OEM-specific information is read from the EEPROM by means of hardware logic, and is transferred to the appropriate registers in the ASIC's PCI/USB core, when the device is turned on by the end consumer. This allows the device to be identified by the computer.

[0011] The registers in the IO blocks which are used to control the rest of the electronics on the device either remain uninitialized, are set to a reset value which has been permanently "burnt in" in the ASIC, or can be set by the hardware logic according to the EEPROM content.

[0012] This known solution ~~has the~~**at least two** drawbacks, which are described in more detail below.

[0013] First, the hardware required for this is relatively complex. The logic needs to be implemented in the ASIC in hard-wired form. Especially with complicated bus protocols, such as I^2C, verification and implementation of the logic (additional gates, transistors, surface area on the silicon die) are complex.

[0014] Secondly, the existing method is not flexible. During the actual chip design phase, it is necessary to define which registers in the ASIC (address) need to be written to (initialized) at what time and with what content (data item) from the EEPROM. The logic described above needs to be designed in line with these requirements.

[0015] ~~The invention is thus based on the object of specifying a method in which~~**Consideration is now being given to ways of improving initialization methods and systems. In particular, attention is directed to procedures for initializing** all registers and modules ~~can be initialized,~~**and to** flexible initialization of the external electronics ~~is possible, ASIC~~**. Desired procedures will simplify ASIC design and** development ~~is simplified,~~ and **allow** different external and internal storage media ~~are~~**to be** supported.

SUMMARY OF THE INVENTION

[0016] ~~The invention achieves the object with a method~~**In accordance with the principles of the invention, and procedures are provided** for initializing programmable systems ~~of the type mentioned at the outset in that after~~**. The inventive procedures may be particularly advantageous for use in a system in which information required for initializing registers, internal modules and external modules is stored in external memory (or internal non-**

volatile storage medium). The initialization information is read from storage in response to turn-on or another event which triggers a fresh start, controlled by a program in an instruction memory~~;~~. **The read** initialization information is transferred from ~~an~~**the** external **and/**or internal non-volatile storage medium to an internal memory~~, in that the~~**. The** initialization information contains initialization data and/or at least one initialization program, ~~in that the registers and modules are initialized under the control of~~**which controls** one or more processor elements or other intelligent building blocks **that are suitably** arranged in the system~~, which, for their part, are controlled by~~**. These processor elements or other intelligent building blocks are in turn configured to control** the initialization ~~program~~**of the registers and modules**.

[0017] In ~~line with the invention~~**operation**, the registers and modules are initialized by one or more processor elements. This **initializing** processor element**(s)** requires a program for execution after the **programming system (or** device**)** has been turned on or after an external restart event for it. The program for starting the initialization phase is contained in an "instruction memory" (bootstrap loader). This program controls the transmission of the initialization information**, for example,** from an external EEPROM to a ~~RAM store~~ **(Random Access Memory (RAM), which may be an** instruction and/or data RAM~~) (RAM = Random Access Memory~~). In this ~~case~~**example**, the initialization information may contain both initialization data and an initialization program. ~~Initialization~~**The initialization** data ~~are~~**may include** identifications (ID), such as the product ID, vendor ID, subsystem vendor ID or a serial number for an Ethernet network card. The initialization program controls the processor element

after the initialization information has been transmitted to the instruction RAM, and implements the initialization of the registers and modules.

[0018] In one refinement of the invention, an integrity check on the initialization information is performed after the transfer, and a program branch is carried out under the control of the result of the integrity check.

[0019] In one embodiment of the invention, upon identification of an incorrect or missing internal or external storage medium, an error routine is executed, which carries out the initialization with standard values, or fully or partially restores the content of the internal or external storage medium.

[0020] ~~It is likewise conceivable for~~**In some applications of the invention,** the information ~~to~~**may** be in the form of an executable macro program and ~~to~~**may** be interpreted by the processor element. Hybrid forms comprising both methods are derivable.

[0021] When the initialization information has been transmitted to the instruction RAM, the data are subjected to an integrity check, for example by ascertaining a checksum. Depending on the result, the processor element either executes the initialization program or macro instructions just transmitted or skips to a routine for handling exceptional cases in the instruction memory. This routine programs the functionally most important registers in the ASIC such that it is at least basically possible to address the device using the respective PC interface. If a missing or incorrect storage medium has been identified in the turn-on phase, a routine for initialization with standard values is likewise executed.

[0022] In a further refinement of the invention, the initialization data are read as standard values from the storage medium, are altered by the processor element, and the altered initialization data are used for initialization.

[0023] The EEPROM stores standard values, for example, for the product ID, vendor ID, subsystem vendor ID ~~or~~, a serial number for an Ethernet network card, ~~inter alia~~**etc**. These values can be used directly for initializing the registers and/or modules. Alternatively, the processor element can alter the standard values or make an alternative selection under the control of an event. From the point of view of error handling, this results in the opportunity to use the support logic to alter or restore the initialization information in the external EEPROM, with it also being possible to recalculate the checksum.

[0024] In one refined form of the invention, the initialization program for the processor element calculates initialization data and uses them for initialization. The processor element can, under the control of the initialization program, calculate initialization data, for example, on the basis of the state of a port or register.

[0025] In one embodiment of the invention, state data for peripheral components and/or internal components are taken as a basis for calculating the initialization data for said components and the data for the internal components. The **execution of the** initialization program~~'s program execution~~ may be ~~in a form~~**arranged** such that an initialization value is chosen or calculated on the basis of states of individual internal or external registers or modules. To this end, in a first step, for example, a register or a port is interrogated and the initialization is

performed on the basis of this result after a skip to a point provided for this purpose in the program execution.

[0026] In one particular embodiment of the invention, the processor element changes to a power-saving mode following initialization. ~~When~~ **After** initialization has taken place, there is the opportunity to put the processor element into a power-saving mode, from which it ~~is~~**can be** reset**, for example** by a signal from a PC or from a peripheral module~~, for example~~.

[0027] In ~~one~~**another** particular embodiment of the invention, the initialization of further processor elements is started and monitored. The **"initial" or "first"** processor element can initialize further processors which are present in the system, which subsequently start their own initialization routines. The termination of the initialization is either reported back to the first processor, or the first processor passes control to another processor.

[0028] In a further embodiment of the invention, adaptation to various storage media is performed. The instruction memory arranged in the ASIC contains a start program for execution after the device has been turned on and implements the reading of the initialization information from the external EEPROM and the transmission to the instruction RAM. This start program contains a routine which identifies the connected storage medium and ensures that the respective necessary transmission protocol is observed.

[0029] In a further refined form of the invention, the initialization program reloads further data and/or program code from a storage medium. When a particular state or event is reached, the processor element can reload further initialization information (program code or state data).

BRIEF DESCRIPTION OF THE DRAWINGS

[0030] **Further features of the invention, its nature, and various advantages will be more apparent from the following detailed description and the accompanying drawings, wherein like reference characters represent like elements throughout, and in which:**

[0031] ~~The invention will be explained in more detail below with reference to an exemplary embodiment. In the associated drawings,~~

~~Figure 1 shows~~**Figure 1 is** a **schematic** circuit arrangement based on the prior art, and

~~Figure 2 shows a circuit arrangement for implementing the inventive method.~~

[0032] **Figure 2 is a schematic circuit arrangement configured for implementing an initialization procedure in accordance with the principles of the present invention.**

[0033] **The following is a list of the reference numerals used in Figures 1 and 2**

1 ASIC

2 Bus interface

3 Personal computer (PC)

4 First bus system

5 ID register

6 I/O interface

7 External electronics

8 Processor elements

9 Instruction memory

10 Instruction RAM

11 Data RAM

12 EEPROM interface

13 Support logic

14 Storage medium (e.g. EEPROM)

15 Second bus system

16 Initialization control

17 Primary function logic

18 Peripheral device

DETAILED DESCRIPTION OF THE INVENTION

[0034] The present invention provides procedures for initializing devices and components of a programming system.

[0035] **FIG. 2 show an exemplary ASIC-based programming system which is suitably configured so that the inventive initialization procedures and methods can be used. FIG. 1 shows, for comparison, a prior art system.**

[0036] ~~To~~**With reference to FIG. 2, to** implement the inventive method, the following components were integrated in an ASIC 1 ~~besides the~~**:** **a** bus interface 2 (known from the prior art), which a first bus system 4 uses to couple to a PC 3 and which contains ID registers 5 for initialization, and the I/O interface 6, which provides the coupling to the external electronics ~~7~~**7. Additionally, the following components are integrated in ASIC 1**: a processor element ~~8,~~**8** (which in the prior art does not necessarily have to be part of the ASIC logic ~~1,~~**1);** an instruction memory 9 (bootstrap loader)~~,~~**;** an instruction RAM ~~10,~~**10;** a data RAM ~~11,~~**11;** an EEPROM interface 12**;** and a support logic unit 13. The external EEPROM 14 **(which may be** known from the prior art**)** is likewise arranged outside the ASIC 1 and is connected thereto via a second bus system 15. The second bus system 15 used may**, for example,** be an I^2C, SPI or Microwire bus. The ASIC 1, the external EEPROM 14 and the external electronics 17 form a peripheral device 18 which can be controlled by the PC 3.

[0037] ~~The instruction~~**Instruction** memory 9 **is** arranged in the ASIC ~~contains~~**to contain** a start program, which is to be executed after the device is turned on, and implements the reading of the initialization information from the external EEPROM 14 and the transmission to the instruction RAM 10. Since the interchange of the bus signals is dependent on the respective external EEPROM 14 and bus system 15 used and the operation is controlled by a program, the adaptation to various bus systems 15 or EEPROM types 14 can be implemented by

programming. It is thus also possible to implement automatic recognition of the connected EEPROM type 14.

[0038] ~~For developing the~~**Exemplary** instruction memory ~~9,~~**9 may be developed using** standard development and debugging tools ~~can be used~~ instead of expensive special **development or debugging** tools.

[0039] The initialization information to be transmitted ~~comprises~~**includes** the initialization data and an initialization program. Following transmission of the initialization information to the instruction RAM 10, an integrity check is performed to ensure error-free transmission of the data.

[0040] If this check establishes, for example, from the checksum, that the transmission was free of errors, the continued program execution takes place using the initialization program which has just been transmitted to the instruction RAM 10. This program performs the actual initialization of the registers 5 and modules using the initialization data. By way of example, this initialization makes the necessary settings in the bus interface 2 in order to allow the communication with the PC 3 and the initialization of the other internal modules and of the I/O interface 6 for controlling the external electronics 7.

[0041] If the integrity check reveals an error, for example, because the checksum is incorrect or even no EEPROM 14 is connected, then it is possible to skip to a routine for handling exceptional cases in the instruction memory 9, said routine programming the functionally most important registers in the ASIC 1 such that the device can be addressed via the respective PC interface 4 at least in principle. This basic setting and the support logic 13 integrated in the ASIC

1 make it possible to repair a faulty device by reprogramming the EEPROM 14 with recalculation of the associated checksum or to perform the initial programming in the production facilities.

[0042] The program-controlled initialization means that it is possible to set all registers 5 or modules which can be addressed directly or indirectly by the processor element 8 and to set state machines which are dependent on said registers or modules. It is furthermore possible to perform the initialization dynamically, that is to say as a function of input values. This means that a display 7 can be used immediately in the start-up phase of the device to indicate whether or not a particular condition is met.

[0043] ~~In addition,~~**It will be understood that the foregoing is only illustrative of the principles of the invention, and that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention. For example, with reference to FIG. 2,** the inventive method allows ~~the~~ peripheral device 18 to act independently in order to "wake up" the PC 3 from an initial state of rest (e.g. Wake-on-~~Lan~~**LAN** or Wake-up for a fax call).

~~List of Reference Symbols~~

~~1 ASIC~~
~~2 Bus interface~~
~~3 Personal computer (PC)~~
~~4 First bus system~~
~~5 ID register~~
~~6 I/O interface~~
~~7 External electronics~~

~~8 Processor elements~~

~~9 Instruction memory~~

~~10 Instruction RAM~~

~~11 Data RAM~~

~~12 EEPROM interface~~

~~13 Support logic~~

~~14 Storage medium (e.g. EEPROM)~~

~~15 Second bus system~~

~~16 Initialization control~~

~~17 Primary function logic~~

~~18 Peripheral device~~

~~Patent Claims:~~

WE CLAIM:

1. A method for initializing **a** programmable ~~systems, in which the~~**system that is characterized in that** information required for initializing registers and internal and/or external modules is ~~contained~~**stored** in ~~an~~**and read from** external memory ~~and is read~~, **the method** particularly **designed** for application in programmable system-on-chip ASIC elements, ~~characterized in that after turn-on or another event which triggers a fresh start, controlled by a program in an instruction memory (9), initialization information is transferred from an external or internal non-volatile storage medium (14) to an internal memory (10), in that the initialization information contains initialization data and/or~~**the method comprising the steps of:**

after turn-on or other event triggering a fresh start,

(a) transferring initialization information from an external or internal non-volatile storage medium to an internal memory under the control of a program in an instruction memory, wherein the initialization information includes at least one initialization program~~, in that the registers (5) and modules are initialized under the control of one or more processor elements (8) or other intelligent building blocks arranged in the system, which, for their part, are controlled by~~ **and/or initialization data;**

(b) initializing the registers and modules under the control of at least one processor element of the programmable system; and

(c) using the initialization program~~,~~ **to control the processor element to perform step(b).**

2. The method ~~as claimed in~~**of** claim 1, **further** characterized in that an integrity check on the initialization information is performed after the transfer, and in that a program branch is carried out under the control of the result of the integrity check.

3. The method ~~as claimed in~~**of** claim 1, **further** characterized in that, upon identification of an incorrect or missing internal or external storage medium, an error routine is executed which carries out the initialization with standard values or fully or partially restores the content of the internal or external storage medium ~~(14)~~.

4. The method ~~as claimed in~~**of** claim 1, **further** characterized in that the initialization data are read as standard values from the storage medium ~~(14), are~~**and** altered by the processor element ~~(8)~~, and the altered initialization data are used for initialization.

5. The method ~~as claimed in~~**of** claim 1, **further** characterized in that the initialization program for the processor element ~~(8)~~ calculates initialization data and uses ~~them~~**the calculated data** for initialization.

6. The method ~~as claimed in~~**of** claim 5, **further** characterized in that state data for peripheral components ~~(7)~~ and/or internal components are taken as a basis for calculating the initialization data for ~~said components and the data for the internal~~**these** components.

7. The method ~~as claimed in~~**of** claim 1, **further** characterized in that the processor element ~~(8)~~ changes to a power-saving mode following initialization.

8. The method ~~as claimed in~~**of** claim 1, **further** characterized in that the initialization of further processor elements ~~(8)~~ is started and monitored.

9. The method ~~as claimed in~~**of** claim 1, **further** characterized in that adaptation to various storage media is performed.

10. The method ~~as claimed in~~**of** claim 1, **further** characterized in that the initialization program reloads further data and/or program code from a storage medium ~~(14)~~.

ABSTRACT OF THE DISCLOSURE

~~The invention relates to a~~**A** method for initializing programmable systems~~,~~ ~~whereby the information~~ **is provided. Information** necessary for initializing registers and internal and/or external components ~~is held in~~**of the programmable system are stored in and read out of** an external memory ~~and read out. The aim of the invention is a method whereby all.~~ **All suitable** registers and components ~~may~~**of the programming system can** be initialized~~, a flexible~~ **by the method providing flexibility in** initialization of ~~the~~ external electronics ~~is possible, the ASIC development is simplified and various types of EEPROM are supported. Said aim is achieved, whereby~~**. The method involves**, after switching on~~,~~ or ~~after another event causing a~~**other** new start~~, a transfer of~~ **event, transferring** initialization information ~~is carried out~~ from an external or internal non-volatile memory to an internal memory~~,~~**. This transfer is** controlled by a program ~~from~~**in** an instruction memory~~, the~~**. The transferred** initialization information ~~comprises~~**includes** initialization data and ~~or at least one~~ **an** initialization program. **In suitably configured programming systems**, ~~the~~ initialization of ~~the register~~**registers** and **other system** components ~~occurs under the control of one or more~~**is controlled by a** processor ~~elements~~ or ~~the~~**other** intelligent ~~components arranged~~**element** in the system~~, which, for their part, are controlled by~~**. This controlling element operates according to** the initialization program. **Implementations of the initialization procedure advantageously simplify ASIC development and support various types of EEPROM.**

Document comparison done by DeltaView on Tuesday, February 22, 2005 1:47:06 PM

Input:	
Document 1	PowerDocs://NY02/512931/1
Document 2	PowerDocs://NY02/512922/1
Rendering set	1-Bold Underline-Strikethrough

Legend:	
Insertion	
~~Deletion~~	
~~Moved from~~	
Moved to	
Style change	
Format change	
~~Moved deletion~~	
Inserted cell	
Deleted cell	
Moved cell	
Split/Merged cell	
Padding cell	

Redline Summary:		
No.	**Change**	**Text**
1-2	Change	"NY02:512162.1 -" changed to "-"
3	Insertion	BAKER BOTTS L.L.P.
4	Insertion	30 ROCKEFELLER PLAZA
5	Insertion	NEW YORK, NEW YORK 10112-4498
6	Insertion	_____
7	Insertion	TO ALL WHOM IT MAY CONCERN:
8	Insertion	Be it known that WE, JENS...an improvement in
9	Insertion	of which the following is a
10	Insertion	SUBSTITUTE SPECIFICATION
11	Insertion	CROSS REFERENCE TO RELATED APPLICATIONS
12-13	Insertion	[0001] This application...their entireties herein.
14	Insertion	FIELD OF THE INVENTION
15-16	Insertion	[0002] The invention...systems for operation.
17	Change	"[0003] The" changed to

		"BACKGROUND OF THE INVENTION[0003] The"
18	Change	"initializing" changed to "invention relates to a method for initializing"
19	Change	"initializing programmable" changed to "initializing processes of programmable"
20-21	Change	"programmable systems, in...registers and" changed to "programmable systems, for...loading of registers and"
22-23	Change	"external modules is contained in an external" changed to "external modules with...stored in an external"
24	Change	"and is read, particularly" changed to "and is read from the...start-up, particularly"
25	Change	"particularly for application in" changed to "particularly in"
26	Change	"in programmable" changed to "in applications that involve programmable"
27	Change	"system-on-chip ASIC elements" changed to "system-on-chip elements"
28	Change	"elements." changed to "elements (e.g.,...Circuits,"ASIC")."
29-30	Change	"made up of" changed to ", particularly in the...are usually made up of"
31-32	Change	"made up of few components" changed to "made up of a limited number of components"
33	Change	"application-specific, integrated" changed to "application-specific integrated"
34-35	Change	"integrated chip (ASIC =...Circuit). This ASIC" changed to "integrated chip or ASIC. This ASIC"
36-37	Change	"Interconnect), Cardbus controller" changed to "Interconnect), Card bus controller"
38-39	Change	", identification" changed to "FirstFirstly, identification"
40-41	Change	"(e.g. USB, Cardbus) or before" changed to "(e.g. USB, Card bus) or before"
42	Change	"usually" changed to "via a usually"

43	Change	"usually serial bus." changed to "usually via a serial bus."
44-45	Change	"case of a PCI/Cardbus, for example," changed to "case of a PCI/Card bus, for example,"
46-47	Change	"drawbacks" changed to "has theat least two drawbacks"
48	Change	"drawbacks which are described" changed to "drawbacks, which are described"
49-50	Change	"all registers" changed to "The invention is thus...all registers"
51-52	Change	"and modules can be...flexible initialization" changed to "and modules and to flexible initialization"
53-54	Change	"electronics is possible, ASIC development" changed to "electronics. Desired...design and development"
55-56	Change	"development is simplified and" changed to "development, and"
57	Change	"and different external" changed to "and allow different external"
58-59	Change	"storage media are supported." changed to "storage media to be supported."
60	Insertion	SUMMARY OF THE INVENTION
61-62	Change	"for initializing" changed to "The invention achieves...for initializing"
63-64	Change	"programmable systems of...in that after turn-on or" changed to "programmable systems. ...response to turn-on or"
65-66	Change	"instruction memory, initialization" changed to "instruction memory. The read initialization"
67-68	Change	"transferred from an external" changed to "transferred from the external"
69	Change	"external or internal" changed to "external and/or internal"
70-71	Change	"internal memory, in that the initialization" changed to "internal memory. The initialization"
72-73	Change	"initialization program,...control of one or more" changed to "initialization program,...controls one

		or more"
74	Change	"building blocks arranged in" changed to "building blocks that are suitably arranged in"
75-76	Change	"in the system, which, for...by the initialization" changed to "in the system. These...the initialization"
77-78	Change	"initialization program." changed to "initialization of the registers and modules."
79-80	Change	", the registers" changed to "line with the inventionoperation, the registers"
81	Change	"elements. This processor element" changed to "elements. This initializing processor element"
82	Change	"processor element requires a" changed to "processor element(s) requires a"
83	Change	"execution after the device" changed to "execution after the programming system (or device"
84	Change	"device has been turned" changed to "device) has been turned"
85	Change	"information from an external" changed to "information, for example, from an external"
86-87	Change	"EEPROM to a RAM store (instruction" changed to "EEPROM to a Random Access...may be an instruction"
88	Change	"and/or data RAM) (RAM = Random Access Memory)." changed to "and/or data RAM."
89-90	Change	". In this case, the initialization" changed to ". In this example, the initialization"
91-92	Change	"initialization program. Initialization data" changed to "initialization program. The initialization data"
93-94	Change	"data are identifications" changed to "data may include identifications"
95	Change	"which carries" changed to ", which carries"
96	Change	"standard values or fully or" changed to "standard values, or fully or"
97-98	Change	"the information" changed to "It is likewise...the information"
99-100	Change	"information to be in the form" changed to "information may be in the form"
101-102	Change	"program and to be interpreted" changed to "program and may be interpreted"
103	Change	"for the product" changed to ", for the product"
104-105	Change	"subsystem vendor ID or a serial number" changed to "subsystem vendor ID, a serial

		number"
106-107	Change	"network card, inter alia. These values" changed to "network card, etc. These values"
108	Change	"on the basis" changed to ", on the basis"
109	Change	"initialization" changed to "execution of the initialization"
110	Change	"initialization program's program execution may be" changed to "initialization program may be"
111-112	Change	"may be in a form such that an" changed to "may be arranged such that an"
113-114	Change	"initialization" changed to "When After initialization"
115-116	Change	"from which it is reset" changed to "from which it can be reset"
117	Change	"reset by a signal" changed to "reset, for example by a signal"
118	Change	"peripheral module, for example." changed to "peripheral module."
119-120	Change	"particular" changed to "oneanother particular"
121	Change	"The processor element" changed to "The "initial" or "first" processor element"
122	Insertion	BRIEF DESCRIPTION OF THE DRAWINGS
123-124	Insertion	[0030] Further features...and in which:
125-126	Change	"a" changed to "The invention will be...1 showsFigure 1 is a"
127	Change	"a circuit arrangement" changed to "a schematic circuit arrangement"
128	Deletion	prior art, and Figure 2...the inventive method.
129-130	Insertion	[0032] Figure 2 is a...the present invention.
131-132	Insertion	[0033] The following is a...used in Figures 1 and 2
133	Insertion	1 ASIC
134	Insertion	2 Bus interface
135	Insertion	3 Personal computer (PC)
136	Insertion	4 First bus system
137	Insertion	5 ID register
138	Insertion	6 I/O interface
139	Insertion	7 External electronics
140	Insertion	8 Processor elements
141	Insertion	9 Instruction memory
142	Insertion	10 Instruction RAM
143	Insertion	11 Data RAM
144	Insertion	12 EEPROM interface
145	Insertion	13 Support logic

146	Insertion	14 Storage medium (e.g. EEPROM)
147	Insertion	15 Second bus system
148	Insertion	16 Initialization control
149	Insertion	17 Primary function logic
150	Insertion	18 Peripheral device
151	Insertion	DETAILED DESCRIPTION OF THE INVENTION
152-154	Insertion	[0034] The present...of a programming system.
155-156	Insertion	[0035] FIG. 2 show an...a prior art system.
157-158	Change	"implement the" changed to "ToWith reference to FIG. 2, to implement the"
159-160	Change	"in an ASIC 1 besides the bus interface" changed to "in an ASIC 1: a bus interface"
161-162	Change	"electronics 7: a processor" changed to "electronics 7. ...in ASIC 1: a processor"
163-164	Change	"processor element 8, which in the" changed to "processor element 8 (which in the"
165-166	Change	"ASIC logic 1, an instruction" changed to "ASIC logic 1); an instruction"
167-168	Change	"(bootstrap loader), an instruction" changed to "(bootstrap loader); an instruction"
169-170	Change	"instruction RAM 10, a data RAM" changed to "instruction RAM 10; a data RAM"
171-172	Change	"a data RAM 11, an EEPROM interface" changed to "a data RAM 11; an EEPROM interface"
173	Change	"interface 12 and a support" changed to "interface 12; and a support"
174	Change	"external EEPROM 14 known from the" changed to "external EEPROM 14 (which may be known from the"
175	Change	"the prior art is likewise" changed to "the prior art) is likewise"
176	Change	"15 used may be an I^2C," changed to "15 used may, for example, be an I^2C,"
177-178	Change	"memory 9" changed to "The instructionInstruction memory 9"
179	Change	"memory 9 arranged in" changed to "memory 9 is arranged in"
180-181	Change	"in the ASIC contains a start program," changed to "in the ASIC to contain a start program,"
182-183	Change	"instruction" changed to "For developing theExemplary instruction"
184-185	Change	"instruction memory 9, standard development" changed to "instruction memory 9 may...standard development"

186	Change	"debugging tools can be used instead of expensive" changed to "debugging tools instead of expensive"
187	Change	"expensive special tools." changed to "expensive special development or debugging tools."
188-189	Change	"the initialization" changed to "comprisesincludes the initialization"
190	Change	"from the checksum," changed to ", from the checksum,"
191	Change	"because the" changed to ", because the"
192-193	Change	"the inventive" changed to "In addition,It will be...to FIG. 2, the inventive"
194	Change	"method allows the peripheral device" changed to "method allows peripheral device"
195-196	Change	"(e.g. Wake-on-Lan or Wake-up" changed to "(e.g. Wake-on-LAN or Wake-up"
197	Deletion	List of Reference Symbols
198	Deletion	1 ASIC
199	Deletion	2 Bus interface
200	Deletion	3 Personal computer (PC)
201	Deletion	4 First bus system
202	Deletion	5 ID register
203	Deletion	6 I/O interface
204	Deletion	7 External electronics
205	Deletion	8 Processor elements
206	Deletion	9 Instruction memory
207	Deletion	10 Instruction RAM
208	Deletion	11 Data RAM
209	Deletion	12 EEPROM interface
210	Deletion	13 Support logic
211	Deletion	14 Storage medium (e.g. EEPROM)
212	Deletion	15 Second bus system
213	Deletion	16 Initialization control
214	Deletion	17 Primary function logic
215	Deletion	18 Peripheral device
216	Deletion	Patent Claims:
217	Insertion	WE CLAIM:
218	Change	"initializing programmable" changed to "initializing a programmable"
219-220	Change	"programmable systems, in which the information" changed to "programmable system that...in that information"
221-222	Change	"modules is contained in" changed to "modules is stored in"
223-224	Change	"in an external memory" changed to "in and read

		from external memory"
225	Change	"external memory and is read," changed to "external memory,"
226	Change	", particularly" changed to ", the method particularly"
227	Change	"particularly for application" changed to "particularly designed for application"
228-229	Change	"ASIC elements,...data and/or" changed to "ASIC elements, the method...comprising the steps of:"
230	Insertion	after turn-on or other...a fresh start,
231	Change	"at least one" changed to "(a) transferring...includes at least one"
232-233	Change	"initialization program,...part, are controlled by" changed to "initialization program...initialization data;"
234	Insertion	(b) initializing the...programmable system; and
235	Change	"the initialization" changed to "(c) using the initialization"
236-237	Change	"initialization program." changed to "initialization program to...to perform step(b)."
238-239	Change	"2. The method as claimed in claim 1," changed to "2. The method of claim 1,"
240	Change	"claim 1, characterized" changed to "claim 1, further characterized"
241-242	Change	"3. The method as claimed in claim 1," changed to "3. The method of claim 1,"
243	Change	"claim 1, characterized" changed to "claim 1, further characterized"
244	Change	"storage medium (14)." changed to "storage medium."
245-246	Change	"4. The method as claimed in claim 1," changed to "4. The method of claim 1,"
247	Change	"claim 1, characterized" changed to "claim 1, further characterized"
248-249	Change	"storage medium (14), are altered by" changed to "storage medium and altered by"
250	Change	"processor element (8), and the altered" changed to "processor element, and the altered"
251-252	Change	"5. The method as claimed in claim 1," changed to "5. The method of claim 1,"
253	Change	"claim 1, characterized" changed to "claim 1, further characterized"
254	Change	"processor element (8) calculates" changed to "processor element calculates"
255-256	Change	"data and uses them for initialization." changed to

		"data and uses the...data for initialization."
257-258	Change	"6. The method as claimed in claim 5," changed to "6. The method of claim 5,"
259	Change	"claim 5, characterized" changed to "claim 5, further characterized"
260	Change	"components (7) and/or internal" changed to "components and/or internal"
261-262	Change	"initialization data for...the internal components." changed to "initialization data for these components."
263-264	Change	"7. The method as claimed in claim 1," changed to "7. The method of claim 1,"
265	Change	"claim 1, characterized" changed to "claim 1, further characterized"
266	Change	"processor element (8) changes to" changed to "processor element changes to"
267-268	Change	"8. The method as claimed in claim 1," changed to "8. The method of claim 1,"
269	Change	"claim 1, characterized" changed to "claim 1, further characterized"
270	Change	"processor elements (8) is started" changed to "processor elements is started"
271-272	Change	"9. The method as claimed in claim 1," changed to "9. The method of claim 1,"
273	Change	"claim 1, characterized" changed to "claim 1, further characterized"
274-275	Change	"10. The method as claimed in claim 1," changed to "10. The method of claim 1,"
276	Change	"claim 1, characterized" changed to "claim 1, further characterized"
277	Change	"storage medium (14)." changed to "storage medium."
278-279	Change	"The invention relates to a method for" changed to "A method for"
280-281	Change	"programmable systems,...necessary for" changed to "programmable systems is...necessary for"
282-283	Change	"components is held in an external" changed to "components of the...read out of an external"
284-285	Change	"external memory and read...all registers and" changed to "external memory. All suitable registers and"
286-287	Change	"components may be initialized" changed to "components of the...can be initialized"
288-289	Change	"initialized, a flexible initialization" changed to "initialized by the method...in initialization"

290	Change	"initialization of the external electronics" changed to "initialization of external electronics"
291-292	Change	"electronics is possible,...whereby, after switching" changed to "electronics. The method...after switching"
293	Change	"switching on, or" changed to "switching on or"
294-295	Change	"or after another event causing a new start" changed to "or other new start"
296-297	Change	"new start, a transfer of initialization" changed to "new start event, transferring initialization"
298	Change	"information is carried out from an external" changed to "information from an external"
299-300	Change	"internal memory, controlled" changed to "internal memory. This transfer is controlled"
301-302	Change	"by a program from an instruction" changed to "by a program in an instruction"
303-304	Change	"instruction memory, the initialization" changed to "instruction memory. The...initialization"
305-306	Change	"information comprises initialization" changed to "information includes initialization"
307-308	Change	"initialization data...least one initialization" changed to "initialization data and an initialization"
309	Change	"initialization program," changed to "initialization program. ...programming systems,"
310	Change	", the initialization" changed to ", initialization"
311-312	Change	"initialization of the register and" changed to "initialization of registers and"
313	Change	"and components" changed to "and other system components"
314-315	Change	"components occurs under...of one or more processor" changed to "components is controlled by a processor"
316	Change	"processor elements or" changed to "processor or"
317-318	Change	"or the intelligent" changed to "or other intelligent"
319-320	Change	"intelligent components arranged in the system" changed to "intelligent element in the system"
321-322	Change	"in the system, which, for...by the initialization" changed to "in the system. This...to the initialization"
323	Insertion	initialization program. ...various types of EEPROM.

Statistics:	
	Count
Insertions	188
Deletions	135
Moved from	0
Moved to	0
Style change	0
Format changed	0
Total changes	323